IEESE International Journal of Science and Technology (IJSTE), Vol. 13 No. 1, March 2024, 14-22
ISSN : 2252-5297

Predicting Laptop Prices Based on Specifications Using Machine Learning Techniques: An Empirical Study

Ridwan Jauhar Kafabihi[1]

[1] Informatics Engineering, Faculty of Science and Technology, Universitas Islam Negeri Maulana Malik Ibrahim, Malang, **Indonesia**
[1]Email: j.kafabihi@gmail.com

Abstract. Essential for companies to formulate competitive pricing strategies and for consumers to make informed decisions and budget appropriately. In this study, we utilized a dataset containing 992 laptops to explore the importance of different features in predicting laptop prices. We implemented XGBoost alongside several other methods, incorporating 15 features such as rating, RAM, brand, type of operating system, processor brand, number of cores, number of threads, primary storage type, primary storage capacity, and more. The results indicate that the XGBoost model achieved an R^2 value of 0.84, the highest among all methods compared. This demonstrates that the XGBoost model has the highest predictive accuracy. Furthermore, the model provides insights into feature importance in price prediction, benefiting both retailers and consumers.

Key Words: *Laptop price prediction, Machine learning, Feature Importance, XGBoost*

1. Introduction

Laptops is a primary need in modern life, laptop used by everyone including, personal use, professional, or even company [1]. The demand for laptop is exploding as technology grow. Because of the variety of brands, spec and prices available in the market, choosing a suitable spec per price has become a challenge for everyone.

With the ever-increasing competition in the laptop market, the ability to predict prices accurately is much needed. A retail shop which predicts prices accurately will have a good advantage in setting interesting price but yet profitable. Whereas consumers who can estimate prices more accurately can get a better deal and get a better price per performance in the device they needed.

There are many factors that comes to play in predicting the price a laptop, these are feature are called selected feature [2]. Machine learning techniques becoming highly effective tools for price prediction Techniques like linear regression, decision trees, and random forests have been used in many studies to predict prices, including laptops. But with recent advancements, more advanced methods like XGBoost are getting a lot of attention.

The purpose of this study is to explore various features importance in predicting laptop prices using different methods, with a primary focus on XGBoost [3]. We use XGBoost for its ability to give accurate predictions while having the feature importance. This study compares the performance of several method with XGBoost to identify the effective model for laptop price prediction.

In this research, we use a dataset consisting of 992 laptop samples with 15 different features. By applying appropriate modelling techniques, we hope to have useful tool for more accurate price predictions in the laptop market[4]. This study not only contributes to academic literature but also provides practical applications for retailers and consumers in understanding the pricing dynamics in the laptop market.

2. Proposed Methodology

In this study we used a quantitative approach method [5] to explore the importance of various features in predicting laptop prices. In Figure 1, The steps to creating this research: data collection, data preprocessing, data splitting, model training, model evaluation, and results analysis.

Subject Category: Computer Science and Informatics
Received: January 10, 2024. Revised: March 01, 2024.

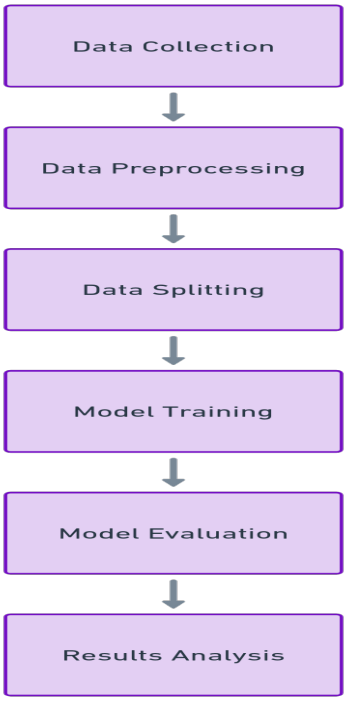

Figure 1. Proposed Method

2.1 Data Collection

The dataset used in this study are 992 laptops, which were sourced from the website kaggle.com. On each of the samples contains a set of features, these features include various spec and characteristics that are important and some of it not relatable for the market value of each laptop.

2.2 Data Preprocessing

The data preprocessing stage directed to process a raw data into a format that suitable for data science [6]. The following steps are involved in this stage: data transformation, data normalization, outlier removal, and feature selection.

2.2.1 Data Transformation

Data transformation is a process to change data from one format to another. In this case, we transform categorical variables into numerical variables using label encoder [6].

```
from sklearn.preprocessing import LabelEncoder
brand_encoder = LabelEncoder()
df['brand'] =brand_encoder.fit_transform(df['brand'].astype(str))
primary_storage_type_encoder = LabelEncoder()
df['primary_storage_type'] = primary_storage_type_encoder.fit_transform(df['primary_storage_type'].astype(str))
processor_brand_encoder = LabelEncoder()
df['processor_brand'] = processor_brand_encoder.fit_transform(df['processor_brand'].astype(str))
gpu_brand_encoder = LabelEncoder()
df['gpu_brand'] = gpu_brand_encoder.fit_transform(df['gpu_brand'].astype(str))
gpu_type_encoder = LabelEncoder()
df['gpu_type'] = gpu_type_encoder.fit_transform(df['gpu_type'].astype(str))
OS_encoder = LabelEncoder()
df['OS'] = OS_encoder.fit_transform(df['OS'].astype(str))
```

Table 1. Features before Encoding Process

BRAND	STORAGE TYPE	PROCESSOR BRAND	GPU BRAND	GPU TYPE	OS
ASUS	SSD	Intel	Nvidia	Dedicated	win
LENOVO	HDD	Intel	Intel	Intergrated	win
APPLE	SSD	Apple	Apple	Apple	mac

Table 2. Features After Encoding Process

BRAND	STORAGE TYPE	PROCESSOR BRAND	GPU BRAND	GPU TYPE	OS
1	1	1	1	1	1
2	2	1	2	2	1
3	1	2	3	3	2

2.2.2 Data Normalization

We used data normalization to make sure that all features have the same scale, which will be important for machine learning algorithms [6]. In this case, we use normalization to the rating feature, where the original scale was 0 to 100 and transformed to a scale of 0 to 1. This prevents the rating feature from dominating the model. The algorithm used for scaling is the Min-Max Scaler from sklearn.preprocessing.

```
scaler = MinMaxScaler()
df['Rating'] = scaler.fit_transform(df[['Rating']])
```

2.2.3 Removing Outliers

Outliers sometimes can lead to high error and extreme bias which will be affecting our model performance this though can be handled by removing outliers. We remove it by identifying and eliminating data that way far from the rest of the data. For example, in our data there are laptops with prices that are extremely high and extremely low compared to other laptops with similar specifications, these laptops can be considered outliers. We can identify outlier using statistical methods Z-score, Removing [7].

2.2.4 Feature Selection

From the data we have the feature is random and have a lot of features. Therefore, we have to select our feature by choosing a subset of the most relevant features from the dataset. This can be done by using techniques like correlation analysis to identify features that have a strong relationship with the price, or using recursive feature elimination to iteratively select the most significant features [8]. But the feature that we used are hand selected using writer's experience from buying a laptop.

Tabel 3. Selected Feature

Fitur	Nama
1	Brand
2	Rating
3	Processor Brand
4	Number of Core
5	Number of Threads
6	Ram Memory
7	Storage Type
8	Storage Capacity
9	GPU Brand
10	GPU Type
11	Touchscreen
12	Display Size
13	Resolution Width
14	Resolution Height
15	Operating System

2.3 Data Splitting

We divided the data into two parts which are training data and testing data, with a ratio of 80:20, meaning 80% for data training and 20% for data testing. This split ensures that the model has sufficient data to learn from while also having a separate dataset for evaluation [9].

2.4 Modelling

In the modelling stage, we run all of the machine learning algorithms to build the laptop price prediction model. The algorithms used include Linear Regression, Decision Tree, Random Forest, and XGBoost.

2.4.1 XGBoost

XGBoost (Extreme Gradient Boosting) is an algorithm popularize by [3]. The advantages of XGBoost are its excellent performance, scalability, and flexibility. Typically, XGBoost is used to solve problems such as classification, regression, and ranking. This is achieved by training multiple weak learners iteratively, which are decision trees. In each iteration, the model corrects errors based on previous predictions [10].XGBoost function comprises a loss function and a regularization term, with the loss function depicted by Equation 1[3].

$$\text{Obj}(\theta) = \sum_{i=1}^{n} = l(y_i, \hat{y}_i) + \sum_{k=1}^{K} = \Omega(f_k) \quad (1)$$

Where:
- l stands for loss function for measuring difference of prediction ŷ and the real value yi.
- Ω used as regularization that will control the O as in complexity of the model to prevent overfitting.
- θ represents model parameters.
- K represent number of trees in the ensemble.

2.4.2 Linear Regression

Relationship between a dependent variable (target) and independent variables (features) assuming that the relationship is linear [11] is called Linear Regression. This relationship can be represented by Equation 2:

$$x = a_0 + a_1 y \quad (2)$$

Where:
x represents dependent variable. y is independent variable.
a 0 is intercept.
a 1 is the coefficient slope.

2.4.3 Random Forest

Random Forest is an ensemble learning method that utilizes a large number of decision trees to generate predictions [12]. Each tree is created from a random sample of data taken with replacement (bootstrap) and randomly selecting features to split each node in the tree. Although Random Forest does not have an explicit equation

$$\hat{y} = \frac{1}{N}\sum_{i=1}^{N} T_i(x) \quad (3)$$

Where:
ŷ is the final prediction.
N is the number of trees in the forest.
Ti(x) is the prediction from the i-th decision tree

2.4.4 Decision Tree

Decision Tree is a machine learning algorithm that generates a model in the form of a tree structure, In the tree, every node symbolizes a decision based on the feature values [13]. Decision Tree divides the data into smaller subsets based on decision rules determined by the dataset's features. At each node n , a decision is made based on the following criteria:

$$Gini = \sum_{i=1}^{C} p_i\,(1 - p_i) \quad (4)$$

Where:
- Gini is the Gini index used to determine the impurity of a node.
- p_i is the proportion of samples that belong to class i.

2.5 Training the Model

Following preprocessing and data splitting, the subsequent stage involves model training. During this phase, we used the 80% of the data to train the model using diverse machine learning techniques. algorithms that have been selected.

2.6 Model Evaluation

The step after training the model using the training data is the next important step which is evaluating the model's performance to understand how well the model can predict laptop prices. Evaluation is done using commonly used evaluation metrics in price prediction contexts, such as Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and R-squared (R2) score [14].

2.6.1 Mean Absolute Error (MAE)

MAE is metric used to measure average absolute error between predicted values $\hat{y}_i$ and actual values yi. A lower MAE value indicates a better model at accurately predicting prices [15].

$$MAE = \left(\frac{1}{n}\right) \sum_{i=1}^{n} |y_i - \hat{y}_i| \quad (5)$$

2.6.2 Mean Squared Error (MSE)

MSE is metric used for measuring the average square difference between actual values and predicted values. A lower MSE value indicates a better model at accurately predicting prices[16].

$$MSE = \Sigma(yi - pi)^2 n \quad (6)$$

2.6.3 Root Mean Squared Error (RMSE)

RMSE is basicly MSE but rooted. RMSE measures the average square root error between predicted values and actual values. A lower RMSE value indicates a better model [15].

$$RMSE = \sqrt{MSE} \quad (7)$$

2.6.4 R-squared (R2)

R2 used to evaluate the regression model performance. R2 show the variance in the dependent variable that is predictable from the independent variables. R2 values range from 0 to 1, with higher values indicating a better model at explaining data variability [16].

$$R^2 = 1 - \sum_{i=}^{n} \frac{(y_i - \hat{y}_i)^2}{(y_i - \hat{y})^2} \quad (8)$$

3. RESULTS AND DISCUSSION

After evaluating several trained models, we collected the evaluation results to compare the performance of each model. The evaluated models include Decision Tree, Random Forest, Linear Regression, and XGBoost. The results are presented in a table and graph format to help analysis.

Tabel 4. Model Evaluation

MODEL	MAE	MSE	RMSE	R2
REGRESI LINIER	29.04	1447.12	38.04	0.75
DECISION TREE	30.67	3020.38	54.95	0.49
RANDOM FOREST	23.38	1084.64	32.93	0.81
XGBOOST	20.90	890.58	29.84	0.84

The table show that, from these models above we can be see that the XGBoost model has the lowest MSE and RMSE values and the highest R2 value among all tested models. This indicates that XGBoost can predict laptop prices more accurately compared to other models.

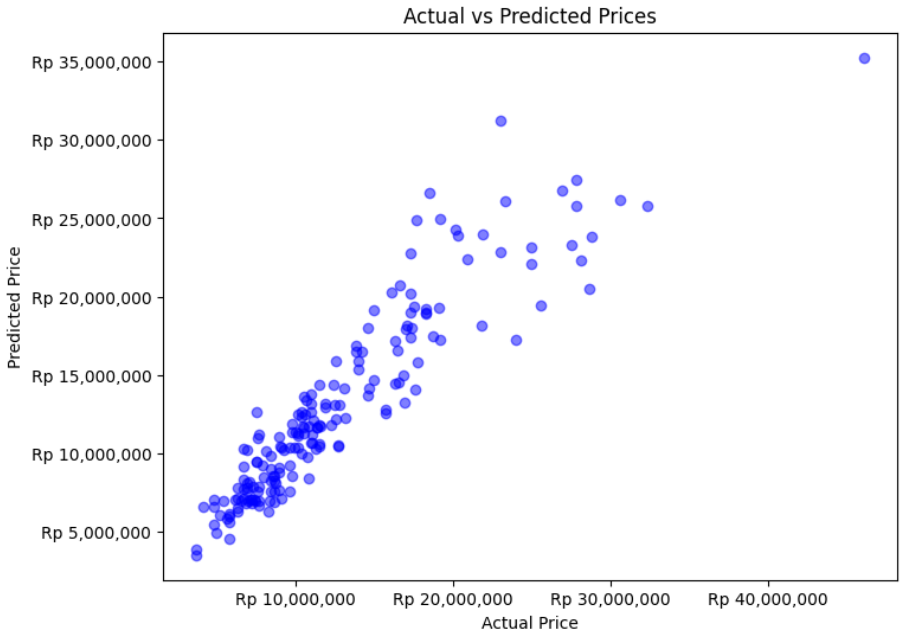

Figure 2. Xgboost Model Performance

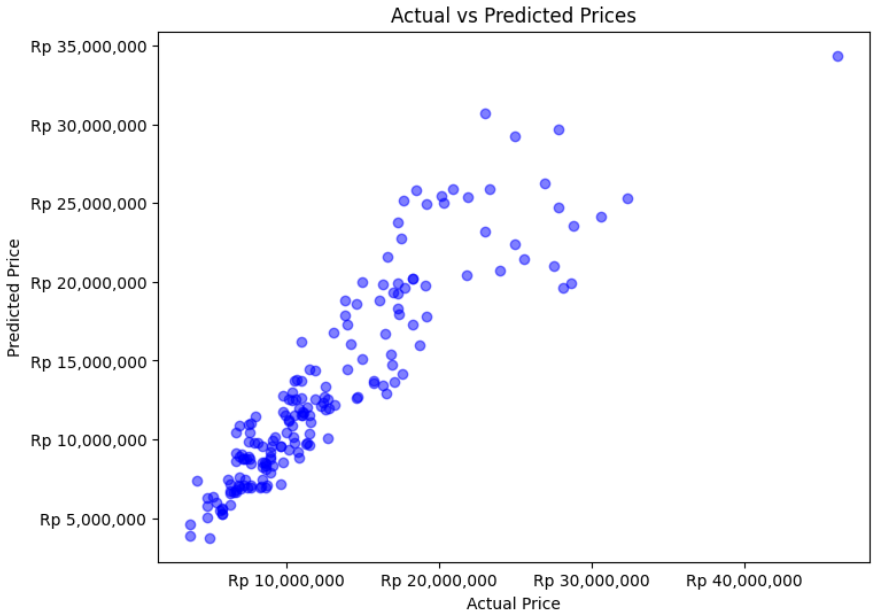

Figure 3. Random Forest Model Performance

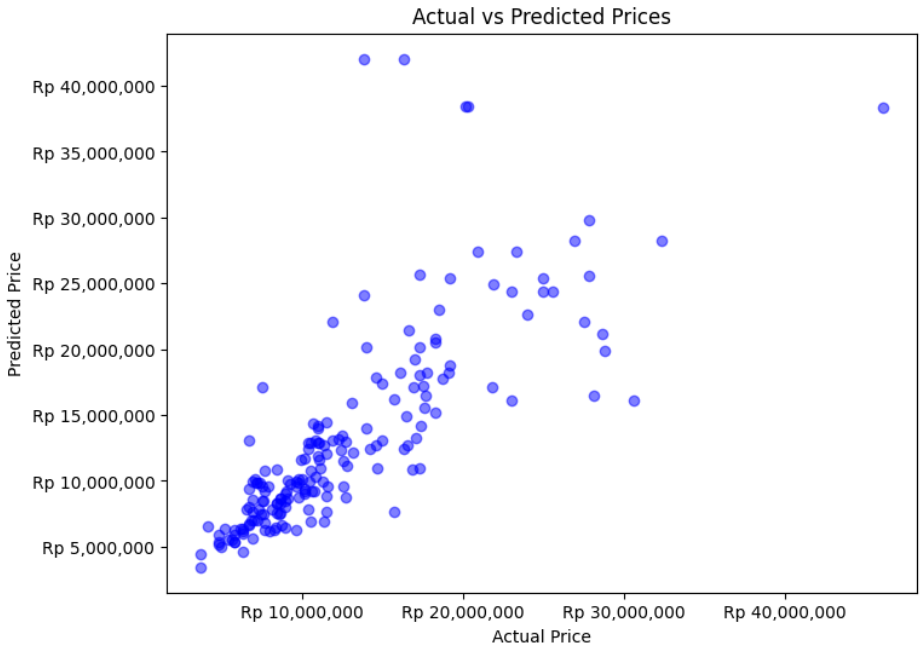

Figure 4. Decision Tree Model Performance

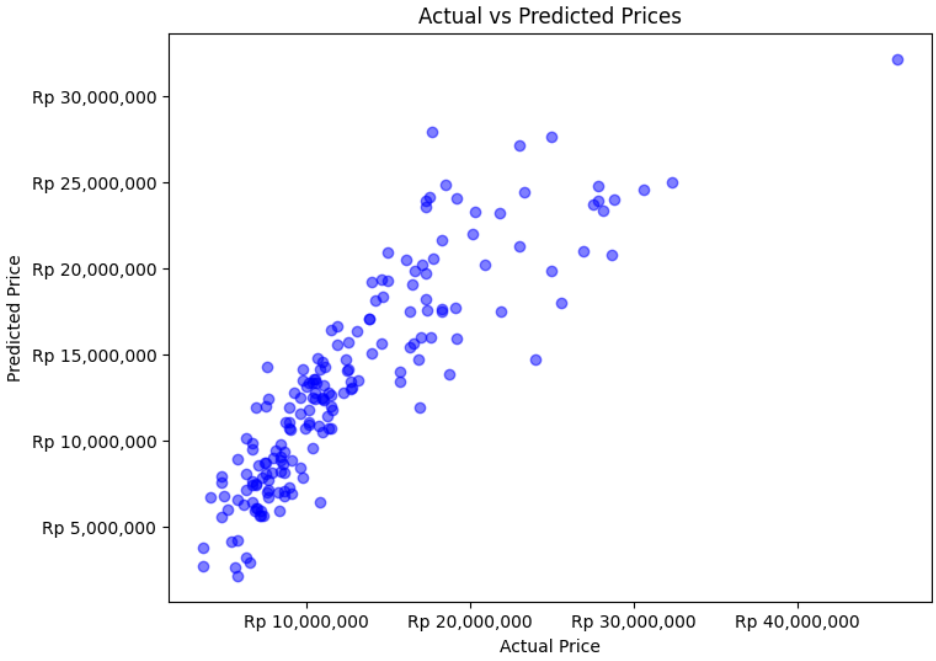

Figure 5. Linear Regression Model Performance

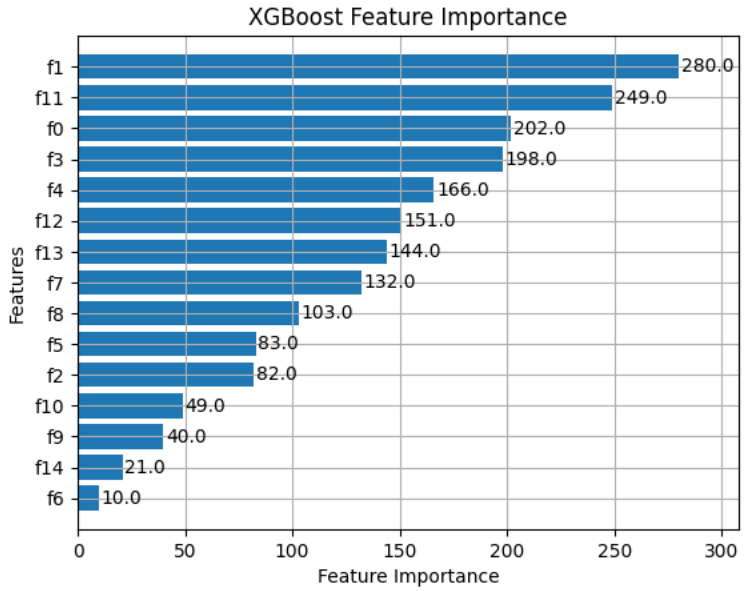

Figure 6. Feature Importance Graph

we can see from the feature importances figure 6, the value of the influence of features used in the classification model was obtained. From the 15 features used in the classification model (in figure start from 0), it was shown that f1 feature, which is the rating feature, have the highest value. The higher the value of a feature, tells that the influence in determining laptop price predictions is higher.

4. Conclusion

In this study, we used several models to predict laptop prices. The results showed that the XGBoost algorithm performed the best, with MAE, MSE, RMSE, and R^2 values of 20.90, 890.58, 29.84, and 0.84, respectively. The XGBoost model also provided insights into feature importance, highlighting the factors that most influence price predictions. Future research could explore larger and more diverse datasets and test price prediction models on other types of electronic products. Additionally, further studies might leverage more advanced machine learning techniques and combine ensemble models to enhance prediction accuracy.

References

[1] D. Wijaya ABA BSI Jakarta Jl Salemba Tengah No, 'Pengaruh Motivasi dan Gaya Hidup Terhadap Keputusan Pembelian', 2017.
[2] J. Ferdinand Sedua, I. Prayanthi, and P. Indomarco Prismatama, 'The Analysis of Factors Influencing Decisions When Buying Laptop', Cogito Smart Journal |, vol. 8, no. 1.
[3] T. Chen and C. Guestrin, 'XGBoost: A Scalable Tree Boosting System', Mar. 2016, doi: 10.1145/2939672.2939785.
[4] P. Tian, 'Research On Laptop Price Predictive Model Based on Linear Regression, Random Forest and Xgboost', 2024.
[5] P. Pada Pendekatan Kualitatif dan Kuantitatif Ardiansyah, Ms. Jailani, S. Negeri, B. Provinsi Jambi, and U. Sulthan Thaha Saifuddin Jambi, 'Teknik Pengumpulan Data Dan Instrumen Penelitian Ilmiah'. [Online]. Available: http://ejournal.yayasanpendidikandzurriyatulquran.id/index.php/ihsan
[6] M. J. Bommarito, 'Preprocessing Data', in Legal Informatics, Cambridge University Press, 2021, pp. 55–60. doi: 10.1017/9781316529683.006.
[7] T. V. Pollet and L. van der Meij, 'To Remove or not to Remove: the Impact of Outlier Handling on Significance Testing in Testosterone Data', Adaptive Human Behavior and Physiology, vol. 3, no. 1, pp. 43–60, Mar. 2017, doi: 10.1007/s40750-016-0050-z.
[8] S. Solorio-Fernández, J. A. Carrasco-Ochoa, and J. Fco. Martínez-Trinidad, 'A review of unsupervised feature selection methods', Artif Intell Rev, vol. 53, no. 2, pp. 907–948, 2020, doi: 10.1007/s10462-019-09682-y.
[9] A. Rácz, D. Bajusz, and K. Héberger, 'Effect of dataset size and train/test split ratios in qsar/qspr multiclass classification', Molecules, vol. 26, no. 4, Feb. 2021, doi: 10.3390/molecules26041111.
[10] J. Dong, Y. Chen, B. Yao, X. Zhang, and N. Zeng, 'A neural network boosting regression model based on

XGBoost', Appl Soft Comput, vol. 125, p. 109067, Aug. 2022, doi: 10.1016/J.ASOC.2022.109067.
[11] D. Maulud and A. M. Abdulazeez, 'A review on linear regression comprehensive in machine learning', Journal of Applied Science and Technology Trends, vol. 1, no. 2, pp. 140–147, 2020.
[12] H. Tyralis, G. Papacharalampous, and A. Langousis, 'A brief review of random forests for water scientists and practitioners and their recent history in water resources', Water (Switzerland), vol. 11, no. 5. MDPI AG, May 01, 2019. doi: 10.3390/w11050910.
[13] D. Shikhman Vladimir and Müller, 'Decision Trees', in Mathematical Foundations of Big Data Analytics, Berlin, Heidelberg: Springer Berlin Heidelberg, 2021, pp. 171–191. doi: 10.1007/978-3-662-62521-7_9.
[14] J. Zhou, A. H. Gandomi, F. Chen, and A. Holzinger, 'Evaluating the quality of machine learning explanations: A survey on methods and metrics', Electronics (Switzerland), vol. 10, no. 5. MDPI AG, pp. 1–19, Mar. 01, 2021. doi: 10.3390/electronics10050593.
[15] T. O. Hodson, 'Root-mean-square error (RMSE) or mean absolute error (MAE): when to use them or not', Geoscientific Model Development, vol. 15, no. 14. Copernicus GmbH, pp. 5481–5487, Jul. 19, 2022. doi: 10.5194/gmd-15-5481-2022.
[16] D. Chicco, M. J. Warrens, and G. Jurman, 'The coefficient of determination R-squared is more informative than SMAPE, MAE, MAPE, MSE and RMSE in regression analysis evaluation', PeerJ Comput Sci, vol. 7, pp. 1–24, 2021, doi: 10.7717/PEERJ-CS.623.